AFRL-RI-RS-TR-2012-035

SECURE, ROBUST AND FLEXIBLE COMPUTING PLATFORM

RUTGERS UNIVERSITY

JANUARY 2012

FINAL TECHNICAL REPORT

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2012-035 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/ /s/

WILLIAM E. STANTON
Work Unit Manager

PAUL ANTONIK, Technical Advisor
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
January 2012	Final Technical Report	August 2010 – August 2011

4. TITLE AND SUBTITLE

SECURE, ROBUST AND FLEXIBLE COMPUTING PLATFORM

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
FA8750-10-1-0221

5c. PROGRAM ELEMENT NUMBER
63788F

6. AUTHOR(S)

Manish Parashar

5d. PROJECT NUMBER
T3TE

5e. TASK NUMBER
RU

5f. WORK UNIT NUMBER
TG

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rutgers University
Center for Autonomic Computing
94 Brett Road
Piscataway, NJ 08854

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITD
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2012-035

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. PA# 88ABW-2012-0415
Date Cleared: 26 January 2012

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goals of this effort were to (1) develop an overall architecture that identifies key services and underlying protocols, (2) investigate the core network and system level requirements for supporting such a platform, (3) prototype key components and services that compose this infrastructure. This document presents progress on this effort and reports on the results.

15. SUBJECT TERMS
Secure Computing, Networks, Content-based Routing, Querying Processes, Distributed Hash Tables

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			WILLIAM E. STANTON
U	U	U	UU	9	19b. TELEPHONE NUMBER (*Include area code*) N/A

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

1.0 SUMMARY

The goals of this effort were to (1) develop an overall architecture that identifies key services and underlying protocols, (2) investigate the core network and system level requirements for supporting such a platform, (3) prototype key components and services that compose this infrastructure. This document presents progress on this effort and reports on the results.

2.0 PROGRESS AGAINST PLANNED OBJECTIVES

The following objectives have been achieved:

1. Exploratory Phase – Infrastructure Layer: Designed infrastructure layer overlay and services, i.e., self-organizing overlay and content-based routing and discovery services. Also explored potential authentication and authorization mechanisms at the overlay level to address requirements of a trusted communication layer.

2. Prototyping Phase: Developed and deployed a prototype of the self-organizing overlay. Evaluated the layer using simulations as well as existing testbeds.

3.0 TECHNICAL ACCOMPLISHMENTS

3.1 Design and Evaluations of Infrastructure Layer Services

The infrastructure layer has two components: a self-organizing overlay layer and content-based routing and discovery services.

3.1.1 Self-organizing Overlay Layer. The self-organizing layer is designed to build a dynamic overlay on top of trusted network elements, and provide abstractions for routing and querying managed information objects across these elements while maintaining the security and trust guarantees provided by the elements. The overlay is designed to be able to robustly handle the addition, deletion, failure or temporary unavailability of these network elements, adding additional levels of application level robustness and availability. Key attributes of the overlay structure include awareness of the physical infrastructure so that it can optimize itself and adapt to changes, security/access control and privacy built on the secure and trusted physical infrastructure by establishing trust between peers in the overlay, assigning access rights to

information in the system and restricting access to the information stored in the system so that only queries with appropriate credentials can discover it, and reliability and fault tolerance at both the overlay and the routine levels. Network elements and end hosts providing resources in the overlay have different roles and accordingly, different access privileges based on their credentials and capabilities. These credential and capabilities are used to manage application level information domains and can implement context and content aware access control.

3.1.2 Content-based Routing/Discovery Services. The routing engine is designed to build on the overlay and support flexible content-based routing and complex querying of managed information objects using partial keywords, wildcards, or ranges. It also guarantees that all resources with information object and data elements that match a query/message will be located. This layer also provides replication and load balancing services, and handles dynamic addition and deletion of data. The layer provides redundancy to prevent data loss. For example, every resource node may maintain the replica of the state of its successor node in the overlay, and reflect changes to this replica whenever its successor notifies it of changes. Correspondingly, it would also notify its predecessor in the overlay of any changes to its state. Consequently, if a resource fails, the predecessor node merges the replica into its state and then makes a replica of its new successor. If a new node joins, the joining node's predecessor updates its replica to reflect the joining node's state, and the successor gives its state information to the joining node. To maintain load balancing, load should be redistributed among the nodes whenever a node joins and leaves.

3.2 System Architecture

The overall architecture is a Distributed Hash Table (DHT), similar to peer-to-peer data lookup systems (Chord, CAN). The key difference is in the way we map information objects to the index space. In existing systems, this is done using consistent hashing [1]. As a result, objects in this case are randomly distributed across nodes without any notion of locality. Our design attempts to preserve locality while mapping the data elements to the index space – all objects in our case are described using a sequence of keywords and these keywords form a multidimensional keyword space where they are the coordinates and the data elements are points in the space. Two data elements are "local" if their keywords are lexicographically close or they have common keywords. Thus, we map data elements to a 1-dimensional index such that indices that are local in

the 1-dimensional index space are also local in this multi-dimensional keyword space. Further, they are mapped to the same node or to nodes that are close together in the overlay network. This mapping is derived using locality-preserving mappings called Space Filling Curves (SFC) [2-4]. Note that locality is not preserved in an absolute sense – documents that match the same query (i.e., share a keyword) can be mapped to disjoint fragments of the index space, called clusters. These clusters may in turn be mapped to multiple nodes, so a query will have to be efficiently routed to these nodes. The infrastructure layer optimizes the querying process using successive refinement and pruning of the queries to reduce the number of clusters generated for a query, and as a result, the number of messages generated. We have proved that for a generic query that defines a polyhedron region in the multi-dimensional space and matches *p%* of the stored data, the percentage of nodes in a typical system with matching data approaches *p%* [5].

3.3 System Operation

The infrastructure layer defines two basic operations: "publish" and "query". Figure 1 illustrates the publishing process: the keywords describing the content of the data element and the SFC-mapping are used to construct the index for the data element, and this index is used to store the element at the appropriate node in the overlay. Figure 2 shows the simple querying process: the query is translated into the corresponding region in the multi-dimensional information space and the relevant clusters of the SFC-based index space, and the appropriate nodes in the overlay are queried.

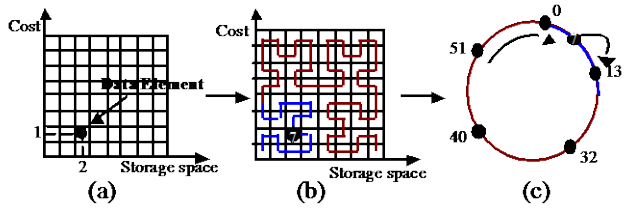

Figure 1. *The process of publishing a data element: (a) the data element (2, 1) is viewed as a point in 2-dimensional space; (b) the data element is mapped to the index 7, using Hilbert SFC; (c) the data element is stored in the overlay (an overlay with 5 nodes and an identifier space from 0 to 2^6-1) at node 13, the successor of the index 7.*

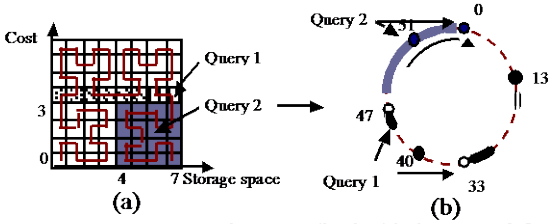

Figure 2. *Processing the query (*, 4): (a) the query defines a rectangular region in the 2-dimensional keyword space, and 3 clusters (3 segments on the SFC curve); (b) the clusters (the solid part of the circle) are stored at nodes 33 and 47, so these nodes will be queried.*

3.4 System Evaluation

In order to evaluate the feasibility of the infrastructure layer and its potential for supporting scalable decentralized content-based discovery, we implemented a simulation using the Hilbert

SFC [2-4] mapping, and the Chord [6] overlay network topology. We used two types of data: uniformly distributed synthetic data; and real data from CiteSeer [7] consisting of HTML files describing scientific articles. In our experiments, we scaled the system size from 10^3 nodes to 10^6 nodes. The number of keys (unique keyword combinations) stored were 10^7 for synthetic data, and $4x10^5$ for real data. 3-dimensional (3D) and 5-dimensional (5D) keyword spaces were evaluated using three types of queries: partial keyword queries, wildcard and partial keyword queries, and range queries, grouped by their coverage (e.g., a query with coverage *p*% matches *p*% of the total data stored in the system). The results of our simulations show that the percentage of nodes with matches decreases as the system size grows, and approaches the percentage of data the query matches, indicating the scalability of the system.

To prove the efficiency of the optimized query engine, we measured the number of clusters generated without and with the optimization. We used queries grouped by coverage into three sets. The optimization greatly reduces the number of clusters generated (e.g., only 0.35% of the clusters need to be generated for the synthetic data, and 0.08% for the real data, for a query with 1% coverage).

We also implemented and deployed a prototype of the infrastructure layer on a cluster of 64 Intel Pentium-4 1.70GHz computers with 512 MB RAM Linux2.4.20-8 and a 100 Mbps Ethernet interconnect. Two sets of queries were tested, their runtime measured, and the results averaged. The first set consisted of keyword queries (no wildcards, ranges, partial keywords), and the second set consisted of wildcard and range queries. Our experiments showed that the runtime of a query grows with the size of the system. As the size of the system grows, the number of intermediary nodes involved in routing the query grows, causing an increase in the runtime. Also, the keyword queries perform better than the wildcard ones. The keyword queries are routed to a single destination, while the wildcard queries are routed to multiple destinations.

4.0 REFRENCES

[1] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web," in *Proceedings of 29th Annual ACM Symposium on Theory of Computing (STOC)*, El Paso, Texas, pp. 654-663, ACM Press, 1997.

[2] T. Bially, "A Class of Dimension Changing Mapping and its Application to Bandwidth Compression," in *Ph.D. dissertation*: Polytechnic Institute of Brooklyn, 1967.

[3] B. Moon, H. Jagadish, C. Faloutsos, and J. Saltz, "Analysis of the Clustering Properties of Hilbert Space-Filling Curve," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13(1), pp. 124-141, 2001.

[4] H. Sagan, *Space-Filling Curves*: Springer-Verlag, 1994.

[5] C. Schmidt and M. Parashar, "Analyzing the Search Characteristics of Space Filling Curves-based Indexing within the Squid Decentralized Data Discovery System," Rutgers University, Piscataway TR-276, 2004.

[6] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proceedings of ACM SIGCOMM*, San Diego, CA, pp. 149-160, ACM Press, 2001.

[7] "CiteSeer webpage," http://citeseer.ist.psu.edu/.